

## **LISTING OF THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Original) A method of converting a software program for a single processor to a software program for a multiprocessor, comprising the steps of: allocating a source file to each processor by an object file unit; and preparing an execute form program for operating software running on a single memory space on the multiprocessor for each processor.

2. (Original) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 1, further comprising the step of: disposing the execute form program mounted on the memory space to be managed by each processor in such a manner that addresses are prevented from being duplicated among the processors.

3. (Original) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 1 or 2, further comprising the steps of: starting exception processing possessed by the processor which is a refer requester to detect that there occurs a refer request to variables arranged on a memory space managed by another processor during running of the execute form program; and sending the refer request to an appropriate processor, wherein the processor which has received the refer request refers to the variables to return refer results to the refer requester, and the processor which is the refer requester emulation-executes a variable refer command from the returned results to return to the next command from the exception processing.

4. (Original) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 1 or 2, further comprising the steps of: starting exception processing possessed by the processor which is a write requester to detect that there occurs a request for write into variables arranged on a memory space managed by another processor during running of the execute form program; and sending the write request to an appropriate processor, wherein the processor which has received the write request writes

the variables, and the processor which is the write requester returns to the next command from the exception processing.

5. (Original) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 1 or 2, further comprising the steps of: starting exception processing possessed by the processor which is a write requester to detect that there occurs a request for write into variables arranged on a memory space managed by another processor during running of the execute form program; and sending the write request to an appropriate processor, wherein the processor which has received the write request writes the variables to return write results to the write requester, and the processor which is the write requester returns to the next command from the exception processing.

6. (Presently Amended) The method of converting the software program for the single processor to the software program for the multiprocessor according to ~~command~~ claim 1 or 2, further comprising the steps of: starting exception processing possessed by the processor which is a call requester to detect that there occurs a call request for functions arranged on a memory space managed by another processor during running of the execute form program; and sending the call request to an appropriate processor, wherein the processor which has received the call request calls the functions to return call results to the call requester, and the processor which is the call requester emulation-executes a function call command from the returned results to return to the next command from the exception processing.

7. (Presently Amended) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 3, ~~4, or 5~~, further comprising: communication between the processors in which communication including processing request transmission and processing result return via the exception processing is possible.

8. (Presently Amended) A cellular phone in which the software program for the multiprocessor converted by the method according to ~~any one of claims 1 to 6~~ claim 1 is installed.

9. (New) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 4, further comprising: communication between the processors in which communication including processing request transmission and processing result return via the exception processing is possible.

10. ( New) The method of converting the software program for the single processor to the software program for the multiprocessor according to claim 5, further comprising: communication between the processors in which communication including processing request transmission and processing result return via the exception processing is possible.

11. (New) A cellular phone in which the software program for the multiprocessor converted by the method according to claim 2 is installed.

12. (New) A cellular phone in which the software program for the multiprocessor converted by the method according to claim 3 is installed.

13. (New) A cellular phone in which the software program for the multiprocessor converted by the method according to claim 4 is installed.

14. (New) A cellular phone in which the software program for the multiprocessor converted by the method according to claim 5 is installed.

15. (New) A cellular phone in which the software program for the multiprocessor converted by the method according to claim 6 is installed.